Bootstrapped Named Entity Recognition for Product Attribute Extraction

Duangmanee (Pew) Putthividhya
eBay Inc.
2065 Hamilton Ave
San Jose, CA 95125
dputthividhya@ebay.com

Junling Hu
eBay Inc.
2065 Hamilton Ave
San Jose, CA 95125
juhu@ebay.com

Abstract

We present a named entity recognition (NER) system for extracting product attributes and values from listing titles. Information extraction from short listing titles present a unique challenge, with the lack of informative context and grammatical structure. In this work, we combine supervised NER with bootstrapping to expand the seed list, and output normalized results. Focusing on listings from eBay's clothing and shoes categories, our bootstrapped NER system is able to identify new brands corresponding to spelling variants and typographical errors of the known brands, as well as identifying novel brands. Among the top 300 new brands predicted, our system achieves 90.33% precision. To output normalized attribute values, we explore several string comparison algorithms and found n-gram substring matching to work well in practice.

1 Introduction

Traditional named entity recognition (NER) task has expanded beyond identifying people, location, and organization to book titles, email addresses, phone numbers, and protein names (Nadeau and Sekine 2007). Recently there has been a surge of interest in extracting product attributes from online data due to the rapid growth of E-Commerce. Current work in this domain focuses on mining product reviews and descriptions from retailer websites. Such text data tend to be long and generate enough context for the target task (Brody and Elhadad 2010; Liu et al. 2005; Popescu and Etzioni 2005). In this paper, we focus on mining short product listing titles, which poses unique challenges.

Short listings are typical in classified ads where each seller is given limited space (in terms of words) to describe the product. On eBay, product listing titles cannot exceed 55 characters in length. Similarly, on Craigslist and newspaper ads, the length of a listing title is restricted. Extracting product attributes from such short titles faces the following challenges:

- Loss of grammatical structure in short listings where many nouns are piled together.
- Typographical errors, abbreviations, and acronyms that must be normalized to the standardized values.
- Lack of contextual information to infer product attribute value.

It can be argued that the use of short listings simplifies the problem of attribute extraction, since short listings can be easily annotated and one can apply supervised learning approach to extract product attributes. However, as the size of the data grows, obtaining labeled training set on the scale of millions of listings becomes very expensive. In such a scenario, incorporating unlabeled examples in a semi-supervised fashion to scale up the solution becomes a necessity rather than a luxury.

We formulate the product attribute extraction problem as a named entity recognition (NER) task and investigate supervised and semi-supervised approaches to this problem. In addition, we have investigated attribute discovery, and normalization to standardized values. We use listings from eBay's clothing and shoes categories and develop an attribute extraction system for 4 attribute types. We have $105,335$ listings from men's clothing category and $72,628$ listings from women's clothing category

Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 1557–1567,
Edinburgh, Scotland, UK, July 27–31, 2011.

on eBay, constituting a dataset of $1,380,337$ word tokens.

In the first part of this work, we outline a supervised learning approach to attribute value extraction where we train a sequential classifier and evaluate the extraction performance on a set of hand-labeled listings. Using maximum entropy and SVM as the base classifier (for classifying the individual word tokens), a hidden Markov model (HMM) is trained on the the probabilistic output of the base classifier, and a sequential label prediction is obtained using a Viterbi decoding. We show a performance comparison of supervised HMM, MaxEnt, SVM, and CRF for this task.

In the second part of our work, to grow our seed list of attributes, we present a bootstrapped algorithm for attribute value discovery and normalization, honing in on one particular attribute (brand). The goal is given an initial list of unambiguous brands, we grow the seed dictionary by discovering context patterns that are often associated with such attribute type. First, we automatically partition data into a training/test set by labeling word tokens in each listing using exact matching to entries in the dictionary. Brand phrases that can be confused with other attributes, e.g. the word *camel* — both a brand and a color — will not be a part of this initial seed list to create the training set. A classifier is then trained to learn context patterns surrounding the known brands from the training set, and is used to discover new brands from the test set.

Finally, for known attribute values, we normalize the results to match to words in our dictionary. Normalizing the variants of a known brand to a single normalized output value is an important aspect of a successful information extraction system. To this end, we investigate several string similarity/distance measures for this task and found that n-gram substring similarity (Kondrak 2005) yields accurate normalization results.

The main contribution of this work is a product attribute extraction system that addresses the unique problems of information extraction from short listing titles. We combine supervised NER with bootstrapping to expand the seed list, and investigate several methods to normalize the extracted results. Our system has been tested on large-scale eBay listing datasets to demonstrate its effectiveness.

2 Related Work

Recent work on product attribute extraction by (Brody and Elhadad 2010) applies a Latent Dirichlet Allocation (LDA) model to identify different aspects of products from user reviews. Similar work is presented in (Liu et al. 2005). Topic models such as LDA groups similar words together by identifying topics (product aspects) from patterns of word-occurrences. Such grouping can discover new aspects of a product such as "portability" (for netbook computers), but it may generate aspects that are vague and not easily interpretable. Indeed, how to refine discovered aspects and clean up words in each aspect remains an open question. The LDA approach also treats documents as bags of words, where important information in word sequences is not taken into account in learning the model.

Our work is most closely related to (Ghani 2006), where a set of product attributes of interests are predefined and a supervised learning method is applied to extract the correct attribute values for each class. Starting out from a small set of training examples, a bootstrapping technique is used to generate more training data from unlabeled data. The main difference to our method lies in how bootstrapping is used. (Ghani 2006) used EM to add more training data from unlabeled data, while in our approach bootstrapping is used to expand the seed list. First, we automatically generate labeled data by matching seed list to unlabeled data. Then, these auto-labeled training set is used to train a classifier to identify new attribute values from a separate set of unlabeled data. Thirdly, newly discovered product attribute values are added back to our seed list. Thus our original classifier for product attribute extraction can be improved through an expanded seed list.

In (Ghani and Jones 2002; Jones 2005), several bootstrapping methods are compared. These methods include self-training, co-EM and EM. All of these approaches are different from ours, as described in detail earlier. In (Probst et al. 2006), a Naive Bayes learner is combined with Co-EM to generate more training data from unlabeled data, and attribute-value pairs are extracted on adjacent words.

The automatic bootstrapping in this paper was inspired by (Pakhomov 2002)—an acronym expansion algorithm for medical text documents. The underlying assumption is that abbreviated forms and their

corresponding expansions occur in similar contexts; consequently, the surrounding context patterns can be used in associating the correct expansion to its acronym.

Our seed list expansion algorithm indeed bears some similarity to the work of (Nadeau el al 2006) and (Nadeau 2007). In (Nadeau el al 2006), automobile brands are learned automatically from web page context. First, a small set of 196 seed brands are extracted together with their associated web page contexts from popular news feed. The web context is subsequently used to extract additional automobile brands, which result in a total of 5701 brands. However, the reported results in (Nadeau el al 2006) have low precision, in some case less than 50%. Eventually their approach needs to rely on rule-based ambiguity resolver to increase the precision. Our system does not rely on manually created rules.

A more NLP-oriented approach is proposed in (Popescu and Etzioni 2005), where noun phrases are extracted from online user reviews. Their system tries to identify product features and user opinions from such noun phrases. A PMI (pointwise mutual information) score is evaluated between each noun phrase and discriminators associated with the product class. The noun-phrase approach does not work well in informal texts. In our case, user-generated short product listings may have many nouns concatenated together without forming a phrase or obeying correct grammatical rules.

Finally, another similar bootstrapping method is presented in (Mintz et al. 2009), where instances of known entity relations (or seed list in our paper) are matched to sentences in a set of Wikipedia articles, and a learning algorithm is trained from the surrounding features of the entities. The trained model is then applied to a test set of Wikipedia articles, and has been reported to be able to discover new instances. In our case, we apply our learned model to a new test set, and discover new brand names from the listings.

The nature of non-grammatical text we face makes our work similar to the NER work on informal texts. (Minkov et al. 2005) proposes an NER system that extracts personal names from emails. The work in (Gruhl et al 2009) identifies song titles from online forums on popular music, where song titles can be very ambiguous. By using real-world constraints such as known song titles, (Gruhl et al 2009) restricts the set of possible entities and are able to obtain reasonable recognition performance.

3 Corpus

The data used in all analysis in this paper is obtained from eBay's clothing and shoes category. Clothing and shoes have been important revenue-generating categories on the eBay site, and a successful attribute extraction system will serve as an invaluable tool for gathering important business and marketing intelligence. For these categories, the attributes that we are interested in are *brand (B)*, *garment type/style (G)*, *size (S)*, and *color (C)*. We gather $105,335$ listings from men's clothing category and $72,628$ listings from women's clothing category, constituting a dataset of $1,380,337$ word tokens. On average, each listing title contains 7.76 words.

A few examples of listings from eBay's clothing and shoes categories are shown in Fig 1. When designing an attribute extraction system to distinguish between the 4 attribute types, we must take into account the fact that individual words alone — without considering context — are ambiguous, as each word can belong to multiple attribute types. To give concrete examples, *inc* is a brand name of women's apparel but many sellers use it as an acronym for inch (brand vs. size). The word *blazer* can be a brand entity or it can be a garment type (brand vs. garment type). In addition, like other real-world user-generated texts, eBay listings are littered with site-specific acronyms, e.g. *BNWT* (brand new with tag), *NIB* (new in box), and abbreviations introduced by individual sellers, e.g. *immac* (immaculate), *trs* (trousers). In designing an information extraction system for our dataset, we need to account for the general as well as specific properties of our dataset.

4 Supervised Named Entity Recognition

In the first part of this work, we adopt a supervised named entity recognition (NER) framework for the attribute extraction problem from eBay listing titles. The goal is to correctly extract attribute values corresponding to the 4 attribute types, from each listing. One key assumption of the supervised learning paradigm is the availability of a labeled training data for training a classifier to distinguish between different classes. We generate our training data in

Figure 1: Example listings and their corresponding labels from the clothing and shoes category.

the following manner. For each listing, we remove extraneous punctuation symbols (*,(,),!,:,;) and tokenize each listing into a sequence of tokens. Given 4 dictionaries of seed values for the 4 attribute types, we match n-gram tokens to the seed values in the dictionaries, and create an initial round of labeled training set, which must then be manually inspected for correctness. In this work, we tagged and manually verified $1,000$ listings randomly sampled from the $105,335$ listings from the men's clothing category, resulting in a total of $7,921$ labeled tokens with $1,521$-word vocabulary. Fig. 1 shows examples of labeled listings, with tags *B* corresponding to brand, *C* for color, *S* for size, *G* for garment type/style, and *NA* for none of the above.

4.1 Classifiers

One of the most popular generative model based classifiers for named entity recognition tasks is Hidden Markov Model (HMM), which explicitly captures temporal statistics in the data by modeling state (label/tag) transitions over time. Discriminative classifiers, which directly model the posterior distribution of class label given features, i.e. SVM (Isozaki and Kazawa 2002) and Maximum Entropy model for NER (Chieu and Ng 2003), have been shown to outperform generative model based classifiers. More recently, Conditional Random Fields (CRF) (Feng and McCallum 2004; McCallum 2003) has been proposed for a sequence labeling problem and has been established by many as the state-of-the-art model for supervised named entity recognition task. In this section, we briefly summarize the pros and cons of each approach.

4.1.1 Hidden Markov Models

A hidden Markov model (HMM) is a probabilistic generative model for sequential data. HMM is characterized by 2 sets of model parameters — emission probabilities which produce the observation variable given the hidden state, and the state transition probability matrix which captures the temporal correlation in the hidden state sequences. Given a set of labeled training sequences as shown in Figure 1, one can train an HMM to model temporal statistics in the observation sequences. In our task, a sequence of word tokens from listing titles are our observations. One simple approach to use HMM is to set a hidden state to correspond to a tag class. In the training phase, since all the tags are given, the hidden states indeed become visible and inference in this model becomes much more simplified. The multinomial parameter for the emission probabilities $p(w|s)$ can be learned with a closed-form update (maximum likelihood estimate). During testing, however, an efficient forward-backward algorithm must be used to infer the most likely tag sequence that accounts for the observation.

One main drawback of HMM is the type of features that it can handle. Like other probabilistic generative models, in order to account for rich, overlapping feature sets, e.g. text formatting features, the correlation structures in the overlapping features must be explicitly modeled. Indeed, in the classic HMM based NER, the simple feature used is the word identity itself, which might not be sufficiently discriminative in distinguishing between different classes. In addition, because of data sparsity (out-of-vocabulary) problem due to the long-tailed distribution of words in natural language, sophisticated unknown word models are generally needed for good performance (Klein et al. 2003).

4.1.2 Maximum Entropy models

The principle of maximum entropy states that among all the distributions that satisfy feature constraints, we should pick the distribution with the highest entropy, since it makes the least assumption about the data and will have better generalization capability to unseen data. Maximum entropy classifier, therefore, is the highest entropy conditional distribution of the class label given features, which has been shown to conveniently take an exponential form. Maximum entropy classifier is thus closely related to logistic regression model.

Position Features:
- Position from the beginning of listing
- Position to the end of listing
Orthographic Features:
- Identity of the current word
- Current word contains a digit
- Current word contains only digits
- Current word is capitalized
- Current word begins with a capitalized letter followed by all non-cap letters.
- Current word is &
- Current word is £
- N-gram substring features of current word ($N = 4, 5, 6$)
Context Features:
- Identity of 2 words before the current word
- Identity of 2 words after the current word
- Previous word is *from*
- Previous word is *by*
- Previous word is *and*
- N-gram substring features of neighboring words ($N = 4, 5, 6$)
Dictionary Features:
- Membership to the 4 dictionaries of attributes
- Exclusive membership to dictionary of brand names
- Exclusive membership to dictionary of garment types
- Exclusive membership to dictionary of sizes
- Exclusive membership to dictionary of colors

Table 1: Feature set used in discriminative classifiers.

MaxEnt classifiers (Ratnaparkhi 1996; Ratnaparkhi 1998) have been applied to various NLP applications. The attraction of the framework lies in the ease with which different information sources used in the modeling process are combined and the good results that are reported with the use of these models. The set of redundant features used for the MaxEnt classifier is the same as those used for the SVM classifier, which we outline in the next section.

4.1.3 Support Vector Machines

Support Vector Machine (SVM) is yet another popular classifier for a supervised NER task. In a binary classification case, SVM finds parameters of a linearly separating hyperplane that best separates data from the 2 classes, in a sense that the margin of separation is maximized. Since only the samples closest to the decision boundary (the so-called support vectors) determine the location of the separating hyperplane, SVM can be trained on very few training examples even for data in a high-dimensional space. For our supervised NER system, we use the following features, as described in detail in Table 1, as input to the discriminative classifiers.

The use of char N-gram (N-gram substring) features was inspired by the work of (Klein et al. 2003), where the introduction of such features has been shown to improve the overall F1 score by over 20%. In (Kanaris et al. 2006), char N-gram features consistently outperform word features in learning effective spam classifiers. Indeed the use of character N-gram features as an input to the classifier subsumes the use of prefix, suffix, and the entire word features. Generally speaking, char N-gram features provide a more robust representation against misspelling since string $s1$ and its spelling variant $s2$ may share many char N-gram substrings in common.

POS and punctuation features are not used in our NER system. This is mainly due to the fact that eBay listing titles are not complete sentences and the output from running a POS tagger through such data can indeed be unreliable. For punctuation features, eBay sellers are known to abuse punctuation marks excessively to draw attention of the potential buyers to click on their listings. In addition, we find that morphological features are less predictive of entity names in eBay listing titles than they are in formal documents. To give a concrete example, capitalization is a good predictor of entity names in traditional NER systems, but on the eBay site, many sellers use all-cap or all-lowercase letters for every word in their titles, bringing into question the discriminative power of widely used features in traditional NER systems.

4.1.4 Viterbi Smoothing

The Viterbi algorithm can be used to smooth the prediction output from SVM or MaxEnt. More specifically, the Viterbi decoder enforces the temporal consistency on the individual label prediction as inferred by the base classifier — MaxEnt or SVM, independently based on the feature representation of each word token. The probabilistic ouput of the base classifier is the observation or evidence, while the temporal consistency is encoded in the empirical state transition probability matrix inferred from the training data. This scenario is analogous to comparing MAP (maximum a priori) estimate with that of ML (maximum likelihood) in that the former incorporates a prior belief when making a final estimate of the parameter values (most likely label sequence predicted by the Viterbi algorithm), while the latter uses only the observation to infer the most likely parameter estimate (independently inferred predicted labels of each word token from the base classifier).

We adopt the approach from the work of (Chieu and Ng 2003), which uses Viterbi to improve the classification results from MaxEnt classifier for NER tasks. Instead of computing the transition probability matrix by recording the frequency of how many times state i at time T transitions to state j at time $T+1$, we simply record that this state i to j transition is admissible. This approach, indeed, divides a set of all label sequences into ones that are admissible and inadmissible, and assign equal probabilities to all the admissible sequences. Such an approach therefore eliminates all the inadmissible sequences of labels (i.e. prohibit the scenario where *-in* sub-tag is followed by *-begin* sub-tag), while allowing the Viterbi algorithm to give more weight to the classification outputs from SVM or MaxEnt.

4.1.5 Conditional Random Field (CRF)

Conditional Random Field, since its conception in the seminal work of (Lafferty et al. 2002), is a discriminative classifier for sequential data that combines the best of both worlds. Like SVM and MaxEnt, CRF is a discriminative classifier that directly models the conditional distribution of the target variable given the observed variable, i.e. no modeling resource is wasted in modeling complex correlation structures in the observation sequences. Like HMM, CRF makes prediction on the label sequence by incorporating the temporal smoothness. Indeed CRF has been established by many as the state-of-the-art supervised named entity recognition system for traditional NER tasks (Feng and McCallum 2004; McCallum 2003), for NER in biomedical texts (Settles 2004), and in various languages besides English, such as Bengali (Ekbal et al. 2008) and Chinese (Mao et al 2008). Various modifications to CRF have recently been introduced to take into account of non-local dependencies (Krishnan and Manning 2006) or broader context beyond training data (Du et al. 2010).

4.2 Experimental Results

In this section, we compare the generative model based and discriminative model classifiers for supervised NER tasks. Given $1,000$ manually tagged listings from the clothing and shoes category in eBay, we adopt a 90-10 split and use 90% of the data for training and 10% for testing. Each listing title is tokenized into a sequence of word tokens, each manually assigned to one of the 5 tags: brand (*B*), size (*S*), color (*C*), garment type (*G*), and none of the above (*NA*). In order to more accurately capture the boundary of multi-token attribute values, we further sub-divide each tag into 2 classes using *-beg* and *-in* sub-tags. This step increases the number of classes that our classifier needs to handle from 5 to 9 classes given as follows: {*B-beg*, *B-in*, *C-beg*, *C-in*, *S-beg*, *S-in*, *G-beg*, *G-in*, and *NA*}.

	SVM	**MaxEnt**	**HMM**	**CRF**
w/o Viterbi	89.05%	87.64%	-	-
w/ Viterbi	**89.47%**	**88.13%**	**83.82**	**93.35%**

Table 2: Classification accuracy (%) on 9-class NER on men's clothing dataset, comparing SVM, MaxEnt, supervised HMM, and CRF.

Table 2 shows a comparison of classification accuracy from 4 classifiers — SVM, MaxEnt, HMM, and CRF. Supervised HMM, with the most simplistic feature, yields the baseline result at 83.82% accuracy. All the discriminative classifiers — CRF, MaxEnt, and SVM — outperform the baseline by HMM, with CRF improving on the baseline performance by the largest margin, concurring to other reports of its state-of-the-art results. Indeed, when using exactly the same set of features as SVM and MaxEnt, the performance of CRF indeed drops to 89.11%, which is on par with that of SVM and MaxEnt. However, when restricting to using dictionary and word identity features, the performance of CRF improves, indicating the importance of feature selection to such model. SVM and MaxEnt yield similar performance with SVM slightly outperforming MaxEnt classifier by 1.6%. The incorporation of temporal smoothness constraint enforced by the Viterbi algorithm slightly improves the label sequence prediction (comparing row 1 and row 2 in Table 2).

The HMM implementation used in our experiments is the Hunpos tagger in (Halacsy et al. 2007), which captures the state transitional probabilities using second-order Markov model. For SVM, we use the popular libSVM package (Chang and Lin 2001) which produces probabilistic output from fitting a sigmoid function to the distances between samples and the separating hyperplane. We use linear kernel in our experiments, although RBF kernel with grid search for optimal parameters yield slightly superior

performance, with a significantly higher computational cost. The MaxEnt implementation used in our experiment is the version available from the NLTK toolkit, with BFGS optimizer. For CRF, we use the linear-chain CRF model available from the Mallet package[1].

5 Bootstrapping for Dictionary Expansion

The supervised learning approach assumes the existence of an annotated set of training data. Often times, training data must be painstakingly marked up and collecting large-scale labeled training examples can be very costly. In recent years, more and more research effort has been focused on how to leverage a vast amount of unlabeled data in a semi-supervised or entirely unsupervised fashion for NER as well as for other similar NLP tasks, e.g. POS tagging, sentence boundary detection, and word sense disambiguation (Riloff 1999; Ghani and Jones 2002; Probst et al. 2006; Brody and Elhadad 2010; Haghighi 2010).

One way to incorporate a vast amount of unlabeled data is to learn a clustering of words that assigns syntactically similar words to the same clusters. Popular clustering algorithms used prevalently in many NER systems are, for example, the combination of distributional and morphological similarity work of (Clark 2003) or the classic N-gram language model based clustering algorithm of (Brown et al. 1992). In such a system, when training an NER classifier, we introduce a word cluster id as an additional feature in the input, with the hope that the model will pick out clusters that are highly indicative of each class. When encountering words that are out-of-vocabulary (OOV) in the test set, if those words are assigned the same cluster membership as some other words in the training set, the cluster feature will fire, allowing for correct classification results to be obtained (Lin and Wu 2009; Faruqui and Pado 2010).

5.1 Growing Seed Dictionary

In this work, we focus on the problem of how to grow the seed dictionary and discovering new brand names from eBay listing data. While the performances of supervised NER classifiers as described in sections 4.1.1-4.1.5 are satisfactory, in practice, however, especially with a small training set size, we often find that the trained model puts too much weight on the dictionary membership feature and new attribute values are not properly detected. In this section, instead of using the seed list of known attribute values as a feature into a classifier, we use the seed values to automatically generate labeled training data. For the specific case of brand discovery, this initial list used to generate training data must contain only names that are unambiguously brands. We hence remove ambiguous names or phrases that belong to multiple attribute types from the list, such as *jumpers*(both a brand name and a garment type), or (ii) *camel* is a short name of brand *Camel active* as well as a color, or (iii) *lrg* is an acronym for a brand as well as an acronym for large which specifies size.

The training/test data is generated by matching N-gram tokens in listing titles to all the entries in the initial brand seed dictionary. Following the convention in (Minkov et al. 2005), we use the following set of 5 tags, (1) one-token entity (B1 tag) (2) first token of a multi-token entity (Bo tag for Brand-open) (3) last token of a multi-token entity (Bc tag for Brand-close) (4) middle token of a multi-token entity (Bi tag for Brand-inside) (5) token that is not part of a brand entity (NA tag). The listings with at least one non-NA tags are put in the training set, and listings that contain only NA tags are in the test set. Similar to the acronym expansion algorithm of (Pakhomov 2002) which learns contexts that associate acronyms to their correct expansions, the intuition behind our work in this section is that the classifier, trained on a labeled training set of known brands, learns context patterns that can discriminate the current word as being a brand (more precisely as part of a brand) from the other attribute types, which are now lumped together as *NA*.

5.2 Experiments

In the first experiment, a set of $72,628$ listings from the women's clothing category is partitioned into a training set of $39,448$ listings and test set of $33,180$ listings based on an initial seed list of known $6,312$ women's apparel brands manually prepared by our fashion experts. The partitioning is done, as described in great detail above, in such a way that known brands in the seed list do not exist in the

[1] http://mallet.cs.umass.edu/

Women's Clothing	Men's Clothing	Garment Type
'monsoon'	henley's	nightshirt
riverislandtop	abercrombie&fitch	cargoshorts
dorothyperkins	lacost	trenchcoat
river islanfd	versace	sweatpants
marks&spencers	sonnetti	cardigans
river islands	supremebeing	boardshorts
river islan	brookhaven	tracksuite
monsoomn	guiness	swimshorts
dorothry perkins	'next'	trouses
principle	suprerdry	microfleece
?river island	henbury	boilersuit
bnwtmonsoon	paul smiths	snopants
marella	ricci	pjs
soulcal	craghopper	jkt

Table 3: Discovered attribute values, ranked order by their confidence scores. (Left) Discovered brands from Women's clothing category. We use 6,312 brands as seed values. (Middle) Discovered brands from Men's clothing category, with 3,499 seed values used. (Right) Discovered garment types (styles) from Men's clothing category, learned from 203 seed values.

test data (using exact string matching criterion). We train a 5-class MaxEnt classifier and adopt the same feature sets as described in Section 4.1.3. During the test phase, the classifier predicts the most likely brand attribute from each listing, where we are only interested in the predictions with confidence scores exceeding a set threshold. We ranked order the predicted brands by their confidence scores (probabilities) and the top 300 unique brands are selected. We manually verify the 300 predicted brands and found that 90.33% of the predicted brands are indeed names of designers or women's apparel stores (true positive), resulting a precision score of 90.33%.

Indeed, the precision score presented above is obtained using an exact matching criterion where partial extraction of a brand is regarded as a miss, i.e. our extractor extracts only *Calvin* when *Calvin Klein* is present in the listing (false positive). The left column of Table 3 shows examples of newly discovered brands from Women's clothing category. Many of these newly discovered brands are indeed misspelled versions of the known brands in the seed dictionary.

The middle column of Table 3 shows a set of Men's clothing brands learned automatically from a similar experiment conducted on a set of $105,335$ listings from Men's clothing category. Using an initial set of $3,499$ known brand seeds, we partition the dataset into a training set of $67,307$ listings and a test set of $38,028$ listings (for later reference we refer to this test set as set A). Based on the top 200 predicted brands, 179 of which are verified as being true positive samples, resulting in 89.5% precision. We carry out a similar experiment to grow the seed dictionary for garment type, and are able to identify the top 60 new garment types. 54 out of 60 are true positive samples, resulting in precision score = 90%. Examples of the newly discovered garment types are shown in Table 3 (right column), where abbreviated forms of garment types such as *jkt* (short for jacket) and *pjs* (short for pajamas) are also discovered through our algorithm.

Seed list	Test set 1	Test set 2
Orig. seeds	83.56%	90.02%
Orig. seed + 200 new brands	92.75%	93.66%

Table 4: NER Accuracy on 2 test sets as the seed dictionary for brands grows. Results shown here are obtained the same Men's clothing category dataset, as used to show the supervised NER results in Table 2.

By adding these newly discovered attributes back to the dictionary, we can now re-evaluate our supervised NER system from section 4 with the grown seed list. To this end, we construct 2 test sets from the same $105,335$ listings of Men's clothing category as used in Section 4. Test set 1 is a set of 500 listings randomly sampled from the $38,028$-listing subset known not to contain any brands in the original brand seed dictionary (set A). As seen in Table 4, an improvement of 9% in accuracy results from the use of the grown seed list. Since this dataset is known to not contain any brands from the original brand seed dictionary, the addition of 200 new brands solely accounts for all the accuracy boost. Test set 2 is constructed slightly differently by randomly sampling 500 listings from the entire $105,335$ listings of Men's clothing category. As seen in Table 4, a smaller improvement of 3.7% is observed.

6 Normalization

With the above described brand discovery algorithm, the newly discovered brands from the test set can be grouped into 2 categories — (i) misspelling, spelling

invariants, abbreviated forms of known brands in the seed list or (ii) novel brands or clothing/shoes designers, which are not members of the original seed list. Normalizing the variants of a known brand to a single normalized output value is an important aspect of our attribute extraction algorithm, as these variants account for over 20% of listings in the eBay clothing and shoes category. When gathering business/marketing intelligence, missing out on 20% of the data could skew the calculation of supply, demand, and pricing metrics, and eventually lead to the wrong policy decision made.

The problem of alternate spellings of names has been addressed in the database community successfully using fuzzy string matching algorithms e.g. Soundex or string edit distance. In this work, since the attribute values are often partially extracted, i.e. a word in a multi-word phrase is extracted, in order to match to the correct normalized value, we must investigate robust substring matching algorithms suitable for partial matching. To this end, we explore 2 string similarity/distance measures for normalizing the extracted attributes. First, we investigate n-gram similarity measures defined as the number of shared character n-grams, i.e. substrings of length n (Kondrak 2005). More specifically, a string similarity measure between $s1$ and $s2$ is defined as the percentage of common substrings of length n (out of all substrings of length n). This similarity measure is quite robust to partial matching, as a two-word phrase can appear out of order while most of the character n-grams, where n = 3, remain virtually unchanged. Certainly, finding the right value of n will greatly impact the matching performance of the algorithm. In our experiment, we find the optimal n for brands to be 3 and 4. Table 5 shows a few examples of normalized outputs as a result of finding the best match for the extracted brand names from among a set of predefined normalized values. When the best matching score falls below a threshold, we declare no match is found and classify the extracted brand as a new brand.

Another distance measure that we explore is the Jaro-Winkler distance. Designed to be more suitable for matching short strings such as people's names, Jaro-Winkler distance is defined based on the number of character transpositions and the number of matching characters. In addition, a prefix scale p parameter is used and can be tuned to weigh more favorably on strings that match from the beginning for a set prefix length. In our experiments with brand normalization, over 50% of the matches from the Jaro-Winkler distance are, however, identified as being incorrect.

Extracted brands	Normalized values
river islands	river island
fruit of loom	fruit of the loom
fruit loom	fruit of the loom
'ralph lauren	ralph lauren
mark & spencer	marks & spencer
yvessaintlaurent	yves saint laurent
yves st laurent	yves saint laurent
combats	combat
'kickers'	kickers
kickers	kickers
armarni	armani
abrecrombie	abercrombie
life & limb	NEW BRAND
oliver baker	NEW BRAND
haines & bonner	NEW BRAND
dehavilland	NEW BRAND
nigel cabourn	NEW BRAND

Table 5: Extracted brands and their corresponding normalized values.

7 Conclusion

In this work, we have described an information extraction system for applications in the domain of inventory/business Intelligence. The goal is given an eBay listing title, our system correctly extracts the defining attributes in order to associate each item to a specific product. We investigate and compare several supervised NER systems — supervised HMM, SVM, MaxEnt, and CRF — and found SVM and MaxEnt with Viterbi decoding to yield the best performance. Focusing on the clothing and shoes categories on eBay's site, we presented a bootstrapped algorithm that can identify new brand names corresponding to (1) spelling invariants or typographical errors of the known brands in the seed list and (2) novel brands or designers. Our attribute extractor correctly discovers new brands with over 90% precision on multiple corpora of listings. To output normalized attribute values, we explore several fuzzy string comparison algorithms and found n-gram substring matching to work well in practice.

8 Acknowledgment

The authors would like to thank Nalini Johnas and Padmanaban Ramasamy for their help in gathering listing data used in all of our experiments.

References

A. Berger, S. Pietra, V. Pietra, *A Maximum Entropy Approach to Natural Language Processing*, ACL 1996.

S. Brody, N. Elhadad, *An Unsupervised Aspect-Sentiment Model for Online Reviews*, HLT-NAACL 2010.

P. Brown, P. deSouza, R. Mercer, V. Della Pietra, J. Lai, Class-based n-gram Models of Natural Language, ACL 1992.

C.-C Chang, C.-J. Lin, LibSVM: A Library for Support Vector Machines (2001).

H. L. Chieu, H. T. Ng, *Named Entity Recognition with a Maximum Entropy Approach*, ACL 2003.

A. Clark, Combining Distributional and Morphological Information for Part of Speech Induction, EACL 2003

G. Demartini, C. S. Firan, M. Georgescu, T. Iofciu, R. Krestel, and W. Nejdl, *An Architecture for Finding Entities on the web*, Latin American Web Congress 2009.

J. Du, Z. Zhang, J. Yan, Y. Cui, and Z. Chen. *Using search session context for named entity recognition in query*. In SIGIR10, Geneva, Switzerland, July 19-23 2010.

Asif Ekbal, Rejwanul Haque, and Sivaji Bandyopadhyay. 2008. *Named entity recognition in Bengali: A conditional random field approach*. In Proceedings of IJCNLP, pages 589594.

M. Faruqui, S. Pado, Training and Evaluating a German Named Entity Recognizer with Semantic Generalization, Proceedings of Konvens 2010, Saarbrucken, Germany.

F. Feng, A. McCallum, *Chinese segmentation and new word detection using conditional random fields*, in COLING 2004.

J. R. Finkel, T. Grenager, and C. Manning, *Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling*, ACL 2005.

J. R. Finkel, C. Manning, *Nested Named Entity Recognition*, EMNLP 2009.

R. Ghani, K. Probst, Y. Liu, M. Krema, A. Fano, *Text Mining for Product Attribute Extraction*, SIGKDD, 2006.

R. Ghani, R. Jones, *A comparison of efficacy and assumptions of bootstrapping algorithms for training information extraction systems*, Workshop on Linguistic Knowledge Acquisition and Representation at the Third International Conference on Language Resources and Evaluation (LREC), 2002.

T. Grenager, D. Klein, and C. D. Manning, *Unsupervised Learning of Field Segmentation Models for Information Extraction*, ACL 2005.

D. Gruhl, M. Nagarajan, J. Pieper, C. Robson, and A. Sheth. *Context and Domain Knowledge Enhanced Entity Spotting In Informal Text*. In Proceedings of the 8th International Semantic Web Conference (ISWC 2009). Springer, 2009.

A. D. Haghighi, *Unsupervised Models of Entity Reference Resolution*, Ph. D. Thesis, University of Calfornia, Berkeley, 2010.

P. Halacsy, A. Kornai, C. Oravecz, *HunPos: an open source trigram tagger*, ACL 2007.

H. Isozaki and H. Kazawa, *Efficient Support Vector Classifiers for Named Entity Recognition*, ACL 2002.

R. Jones, *Learning to Extract Entities from Labeled and Unlabeled Text*, PhD Thesis, 2005.

I. Kanaris, K. Kanaris, I. Houvardas, E. Stamatatos, *Words vs. Character N-grams for Anti-spam Filtering*, International Journal on Artificial Intelligence Tools, 2006.

D. Klein, J. Smarr, H. Nguyen, C. Manning, *Named Entity Recognition with Character-level Models*, CoNLL 2003.

R. Koeling, *Chunking with Maximum Entropy Models*, Proc. of CoNLL-2000.

G. Kondrak, *N-Gram Similarity and Distance*, SPIRE 2005.

V. Krishnan and C. D. Manning, *An effective two-stage model for exploiting non-local dependencies in named entity recognition*, in ACL-COLING, 2006.

T. Kudo, Y. Matsumoto, *Chunking with Support Vector Machines*, ACL 2001.

J. Lafferty, A. McCallum, F. Pereira, *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*, ICML 2002.

V. I. Levenshtein, *Binary code capable of correcting deletions, insertions, and reversals.* Phs. Dokl., 6:707-710.

D. Lin, X. Wu, *Phrase Clustering for Discriminative Learning*, ACL 2009.

B. Liu, M. Hu, and J. Cheng, *Opinion Observer: Analyzing and Comparing Opinions on the Web*, WWW 2005.

Xinnian Mao, Saike He, Sencheng Bao, Yuan Dong, and Haila Wang, *Chinese Word Segmentation and Named Entity Recognition Based on Conditional Random Fields*, Sixth SIGHAN Workshop on Chinese Language Processing, 2008

A. McCallum, *Efficiently Inducing Features of Conditional Random Fields*, UAI 2003.

A. McCallum, D. Jensen, *A Note on Unification of Information Extraction and Data Mining using Conditional-Probability, Relational Models*, Proceedings of IJCAI-2003 on Learning Statistical Models from Relational Data, 2003.

J. F. McCarthy, *A Trainable Approach to Coreference Resolution for Information Extraction*, Ph. D. Thesis, University of Massachusetts at Amherst, 1996.

E. Minkov, R. C. Wang, and W. W. Cohen, *Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text*, ACL 2005.

Mike Mintz, Steven Bills, Rion Snow, Daniel Jurafsky. 2009. *Distant Supervision for Relation Extraction without Labeled Data*, In Proceedings of ACL/AFNLP 2009.

S. Moghaddam, M. Ester, *Opinion Digger: An Unsupervised Opinion Miner from Unstructured Product Reviews*, CIKM 2010

David Nadeau, P. Turney, S.Matwin, *Unsupervised Named Entity Recognition: Generating Gazetteers and Resolving Ambiguity*. In Proc. Canadian Conference on Artificial Intelligence, 2006.

David Nadeau and Satoshi Sekine. *A survey of named entity recognition and classification*. Linguisticae Investigationes, 30(1):326, 2007.

Nadeau, D., *Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision*, PhD thesis, University of Ottawa, 2007.

S. Pakhomov, *Semi-supervised Maximum Entropy Based Approach to Acronym and Abbreviation Normalization in Medical Texts*, ACL 2002.

A.-M. Popescu, O. Etzioni, *Extracting Product Features and Opinions from Reviews*, EMNLP 2005.

K. Probst, R. Ghani, M. Krema, A. Fano, *Semi-Supervised Learning to Extract Attribute-Value Pairs from Product Descriptions on the Web*, ECML 2006.

V. Punyakanok, D. Roth, *The use of classifiers in sequential inference*, NIPS 2001.

H. Raghavan, J. Allan, Matching Inconsistently Spelled Names in Automatic Speech Recognizer Output for Information Retrieval, HLT-EMNLP 2005.

A. Ratnaparkhi, *A Maximum Entropy Part of Speech Tagger.* In EMNLP 1996.

A. Ratnaparkhi, *Maximum Entropy Models forNatural Language Ambiguity Resolution*, Ph. D. Thesis, University of Pennsylvania.

E. Riloff, R. Jones, *Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping*, AAAI 1999.

Settles, B. (2004), *Biomedical named entity recognition using conditional random fields and rich feature sets*, in Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA), 2004, Geneva, Switzerland.

W. M. Soon, H. T. Ng, D. Chung, Y. Lim, *A machine learning approach to coreference resolution of noun phrases*, Computational Linguistics, 27(4): 521-544, 2001.

H. Wallach, *Efficient Training of Conditional Random Fields*, M. Sc. Thesis, Division of Informatics, University of Edinburgh, 2002.

D. Wu, W. S. Lee, N. Ye, and H. L. Chieu, *Domain adaptive bootstrapping for named entity recognition*, EMNLP 2009.

Y. Zhao, B. Qin, S. Hu, T. Liu, *Generalizing Syntactic Structures for Product Attribute Candidate Extraction*, ACL 2010